5

10

METHOD AND APPARATUS FOR ASSIGNING UNIQUE DEVICE IDENTIFIERS ACROSS A DISTRIBUTED COMPUTING SYSTEM

15   **Inventor(s):** Siamak Nazari

20   **Related Application**

The application hereby claims priority under 35 U.S.C. § 119 to Provisional Patent Application No. 60/160,993 filed on October 21, 1999.

BACKGROUND

25

Field of the Invention

The present invention relates to operating systems for computers. More specifically, the present invention relates to a method and an apparatus for assigning device identifiers that are unique across a distributed computing system.

30

1

Related Art

Operating systems for computers often facilitate accessing devices, such as disk drives and printers, through a pathname as well as an integer device identifier. For example, a disk drive may be referenced through a pathname, such as "/dev/rdsk/c0t0d0s0," as well as through an integer device identifier, such as "5064." This integer device identifier can be used by programmers and system administrators to access the device, and can also be used by device drivers to control the device. Note that if two device identifiers have the same value they are assumed to refer to the same device.

The use of device identifiers becomes complicated in a distributed computing system including multiple computing nodes coupled together by a network. Such a distributed computing system is typically controlled by a distributed operating system, which enables the multiple computing nodes to communicate with each other, and enables the computing nodes to share resources, such as devices.

Producing a unique pathname for a device in a distributed computing system is typically accomplished by inserting a name for the computing node into the pathname for the device. For example, the pathname "dev/node1/rdsk/c0t0d0s0" refers to a device located on node1.

However, assigning unique device numbers cannot be accomplished in the same way, because existing distributed operating systems provide no mechanism for combining node identifiers into device identifiers. Hence, local computing nodes typically assign their own local device identifiers. Consequently, the same identifier can potentially be assigned to different devices by different local nodes in the distributed computing system. It is possible to solve this problem by providing a mechanism to translate between local and global device identifiers.

2

However, the process of maintaining such a system is cumbersome and inefficient.

What is needed is a method and an apparatus for assigning device identifiers that are unique across a distributed computing system.

5

SUMMARY

One embodiment of the present invention provides a system that facilitates assigning an identifier to a device within a distributed computing system, wherein the identifier is unique across the distributed computing system. The system

10   operates by detecting the presence of the device within a local computer system within the distributed computing system. If an identifier has not been assigned to the device, the system assigns an identifier to the device by attempting to retrieve the identifier from a local pool of device identifiers within the local computer system. If the local pool is empty, the system retrieves at least one additional

15   identifier for the local pool from a global allocator for device identifiers located within the distributed computing system. Next, the system assigns the retrieved identifier to the device so that the identifier can be used to reference the device.

In one embodiment of the present invention, the identifier includes a device major number that specifies a device driver to be used to access the device,

20   and a device minor number that identifies the device to be accessed by the device driver. This device minor number includes an instance number that uniquely identifies an instance of the device, and a unit number that identifies an independently addressable sub-unit within the device. In a variation on this embodiment, in attempting to retrieve the identifier from the local pool, the

25   system attempts to retrieve the instance number from the local pool. The system combines this instance number with the device major number and the unit number to produce the identifier.

3

In one embodiment of the present invention, if the global allocator is inaccessible, the system assigns a provisional identifier from the local computer system. In a variation on this embodiment, when the global allocator later becomes accessible, the system communicates the provisional identifier to the global allocator. If the global allocator approves the provisional identifier, the system records the provisional identifier as a permanent device identifier. If the global allocator rejects the provisional identifier, the system assigns a new identifier to the device from the global allocator.

In one embodiment of the present invention, retrieving at least one additional identifier from the global allocator involves retrieving a block of identifiers from the global allocator.

In one embodiment of the present invention, the device can include a disk drive, a tape drive, an I/O device or a networking device.
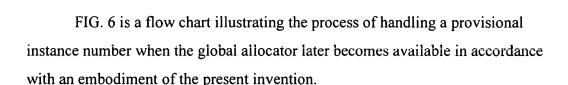
BRIEF DESCRIPTION OF THE FIGURES

FIG. 1 illustrates a distributed computing system in accordance with an embodiment of the present invention.

FIG. 2 illustrates the structure of a device identifier in accordance with an embodiment of the present invention.

FIG. 3 illustrates the structure of a device table in accordance with an embodiment of the present invention.

FIG. 4 illustrates the use of a device identifier in accordance with an embodiment of the present invention.

FIG. 5 is a flow chart illustrating the process of assigning a device identifier in accordance with an embodiment of the present invention.

4

FIG. 6 is a flow chart illustrating the process of handling a provisional instance number when the global allocator later becomes available in accordance with an embodiment of the present invention.

5   **DETAILED DESCRIPTION**

The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed embodiments will be readily apparent to those skilled in the art, and the general

10   principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present invention. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

15   The data structures and code described in this detailed description are typically stored on a computer readable storage medium, which may be any device or medium that can store code and/or data for use by a computer system. This includes, but is not limited to, magnetic and optical storage devices such as disk drives, magnetic tape, CDs (compact discs) and DVDs (digital versatile discs or

20   digital video discs), and computer instruction signals embodied in a transmission medium (with or without a carrier wave upon which the signals are modulated). For example, the transmission medium may include a communications network, such as the Internet.

25   **Distributed Computing System**

FIG. 1 illustrates a distributed computing system 100 in accordance with an embodiment of the present invention. Distributed computing system 100 is

5

comprised of a number of local computer systems 104-107, which are coupled together through a network 102. Computer systems 104-107 can generally include any type of computer system, including, but not limited to, computer systems based on a microprocessor, a mainframe computer, a digital signal

5   processor, a portable computing device, a personal organizer, a device controller, and a computational engine within an appliance. Network 102 can generally include any type of wire or wireless communication channel capable of coupling together computing nodes. This includes, but is not limited to, a local area network, a wide area network, or a combination of networks.

10   Distributed computing system 100 additionally includes global allocator 120, which is a server that assigns device instance numbers (identifiers) to local computer systems 104-107 from a global pool of instance numbers 122. Global allocator 120 can generally include any mechanism for servicing requests from a client for computational and/or data storage resources. Note that global allocator

15   120 can reside on one of local computer systems 104-107 or, alternatively, can reside on a dedicated computing node that is independent of local computer systems 104-107.

Computer systems 104-107 are coupled to a number of local devices 130-136. More specifically, computer system 104 is coupled to devices 130-131;

20   computer system 105 is coupled to device 132; computer system 106 is coupled to devices 133-134; and computer system 106 is coupled to devices 135-136. Devices 130-136 can generally include any type of physical device that is coupled to a computer system and is controlled by a device driver. For example, devices 130-136 can include data storage devices, such as disk drives and magnetic tape.

25   Devices 130-136 can additionally include I/O devices, such as a display or a keyboard. Devices 130-136 can also include networking devices, such as a network interface controller (NIC). Devices 130-136 can also include

6

computational devices, such as a graphics accelerator.

During operation, device instance numbers (identifiers) are assigned to local computer systems 104-107 from global allocator 120. Local computer systems 104-107 then allocates these device identifiers to devices 130-136, so that

5   devices 130-136 can be accessed through these identifiers.

For example, referring the FIG. 1, global pool of instance numbers 122 initially starts out with instance numbers 1-100. Some of these instance numbers 1-100 are then allocated to local computer systems 104-107 so that: local computer system 104 gets instance numbers 1-10; local computer system 105 gets

10   instance numbers 11-20; local computer system 106 gets instance numbers 21-30; and local computer system 107 gets instance numbers 31-40. At this point, global pool instance numbers 122 contains instance numbers 41-100.

Some of these allocated global instance numbers are then assigned to devices, so that: instance numbers 1 and 2 are assigned to devices 130 and 131;

15   instance number 11 is assigned to device 132; instance numbers 21 and 22 are assigned to devices 133 and 134; and instance numbers 31 and 32 are assigned to devices 135 and 136.

At this point, local computer system 104 has instance number 3-10 available to be allocated; local computer system 105 has instance number 12-20

20   available to be allocated; local computer system 106 has instance number 23-30 available to be allocated; and local computer system 107 has instance number 33-40 available to be allocated.

Device Identifier

25   FIG. 2 illustrates the structure of a device identifier 200 in accordance with an embodiment of the present invention. Device identifier 200 is comprised of a device major number 202 and a device minor number 204. In one embodiment of

7

the present invention, device major number 202 is 14 bits in length and device minor number 204 is 18 bits in length. Device major number 202 specifies the type of device that is being referenced, and thereby specifies the device driver that is used to control the device.

5   Device minor number 202 identifies a specific device that is to be controlled by the device driver. Device minor number 202 is further divided into an instance number 206 and a unit number 208. Instance number 206 specifies a specific instance of a device, and unit number 208 specifies an independently addressable sub-unit of the specific instance. For example, a disk drive may be

10   comprised of multiple slices, each one of which is referenced through a different unit number. In another example, a tape drive may include multiple unit numbers for different access modes, such as read-only or read-write.

Note that the terms "device identifier" and "device instance number" are used interchangeably throughout this specification because the device instance

15   number uniquely identifies the device across the distributed computing system.

Device Table

FIG. 3 illustrates the structure of a device table 300 in accordance with an embodiment of the present invention. Device table 300 contains a pathname and a

20   device identifier for each device that is presently configured within a local computer system. Note that each local computer system 104-107 includes its own device table 300. In some distributed computing systems, there exists a system-wide device table containing pathnames and identifiers for all devices in the distributed computing system.

25

8

Use of Device Identifier

FIG. 4 illustrates the use of a device identifier 200 in accordance with an embodiment of the present invention. An application 412 that is operating out of user space 400 within a local computer system makes a system call 410 that

5   includes device identifier 200. This system call 410 is routed to a device driver 403 within kernel space 401 based upon the device major number 202 of device identifier 200.

Note that kernel space 401 includes a number of device drivers 402-404, which can be used to communicate with a number of devices 405-409. More

10   specifically, device driver 402 communicates with devices 405-406, device driver 403 communicates with device 407, and device driver 404 communicates with devices 408-409.

Process of Assigning Device Identifier

15   FIG. 5 is a flow chart illustrating the process of assigning a device identifier in accordance with an embodiment of the present invention. The assignment process typically starts when a device comes on line within a computer system. The system first detects the presence of the device (step 502). Next, the system determines if an identifier has already been assigned to the

20   device (step 504). If so, the process is complete.

Otherwise, the system examines a local pool of instance numbers (identifiers) to determine whether the local pool is empty (step 506).

If the local pool is empty, the system determines if global allocator 120 for identifiers is accessible (step 508). Note that global allocator 120 is often

25   temporarily inaccessible because the assignment process often takes place during system initialization. Hence, global allocator 120 may not yet be initialized, or

9

reachable from the local computer system, when the assignment process takes place.

If global allocator 120 is not accessible, the system gets a provisional instance number from the local computer system (step 510). This provisional

5   instance number can either be generated on-the-fly or, alternatively, can be retrieved from a data structure within the local computer system.

If global allocator 120 is accessible, the system retrieves a block of instance numbers from global allocator 120 for the local pool of identifiers on the local computer system (step 512).

10   Next, if the local pool is not empty in step 506, or if a new block of identifiers has been retrieved in step 512, the system retrieves a device instance number (identifier) from the local pool (step 514).

Next, using either the provisional instance number from step 510 or the newly retrieved instance number from step 514, the system produces an identifier

15   for the device by combining the instance number with a device major number and a unit number as is described in more detail above with reference to FIG. 2.

Finally, this identifier is assigned to the device (step 516). However, note that an identifier constructed from a provisional instance number may later change if the provisional instance number is later determined to be invalid.

20   FIG. 6 is a flow chart illustrating the process of handling a provisional instance number when global allocator 120 later becomes available in accordance with an embodiment of the present invention.

The system starts by detecting the presence of global allocator 120 (step 602). This can be accomplished through a monitoring process that periodically

25   attempts to communicate with global allocator 120.

Next, the system communicates the provisional instance number to global allocator 120 (step 604).

10

Upon receiving the provisional instance number, global allocator 120 compares it with available instance numbers in the global pool of instance numbers 122. If the provisional instance number is one of the available instance numbers, global allocator 120 removes the provisional instance number from the global pool of instance numbers 122, and sends an approval to the local computer system. Otherwise, global allocator 120 sends a disapproval to the local computer system along with an additional block of instance numbers.

Next, the local computer system determines whether global allocator 120 approves of the provisional instance number (step 606). If not, the system receives a new block if instance numbers from global allocator 120 and retrieves an instance number from the new block of instance numbers (step 608). The system then constructs a device identifier from the newly retrieved instance number, and then records the device identifier as a permanent identifier for the device (step 610).

If global allocator 120 approves of the instance number, the system records the identifier constructed from the provisional instance number as the permanent device identifier (step 612). At this point, the identifier assignment process is complete. The device can subsequently be accessed through the device number, and no local-to-global translation for device identifiers is required.

The foregoing descriptions of embodiments of the invention have been presented for purposes of illustration and description only. They are not intended to be exhaustive or to limit the present invention to the forms disclosed. Accordingly, many modifications and variations will be apparent to practitioners skilled in the art. Additionally, the above disclosure is not intended to limit the present invention. The scope of the present invention is defined by the appended claims.

11